Encoder Decoder

→ **Simple Working**

Input Sentence → Encoder (Convert words to array of numbers) → Context Vector [0.1, 0.3, -0.3, 0.6, 0.1] → Decoder → output

Decoder generates the output word by word and keeps feeding the previous word into decoder again
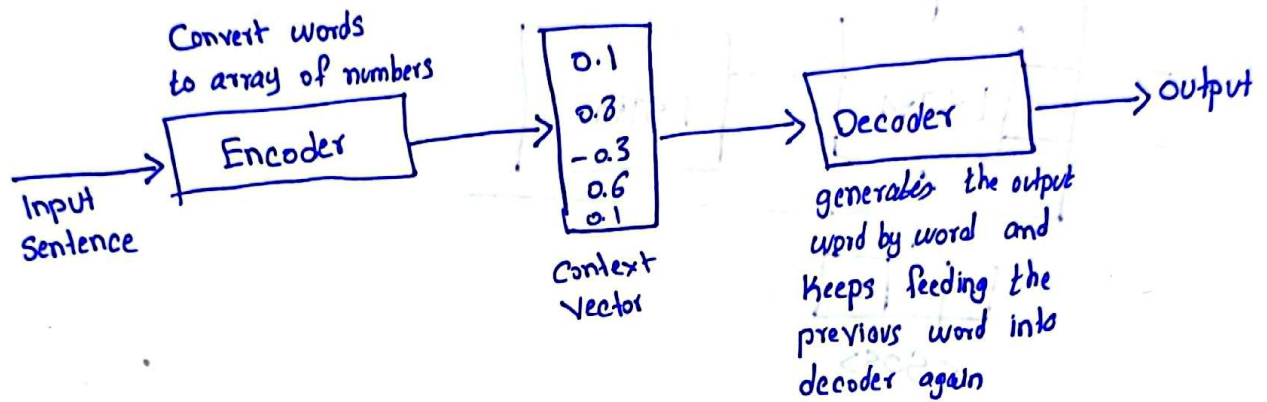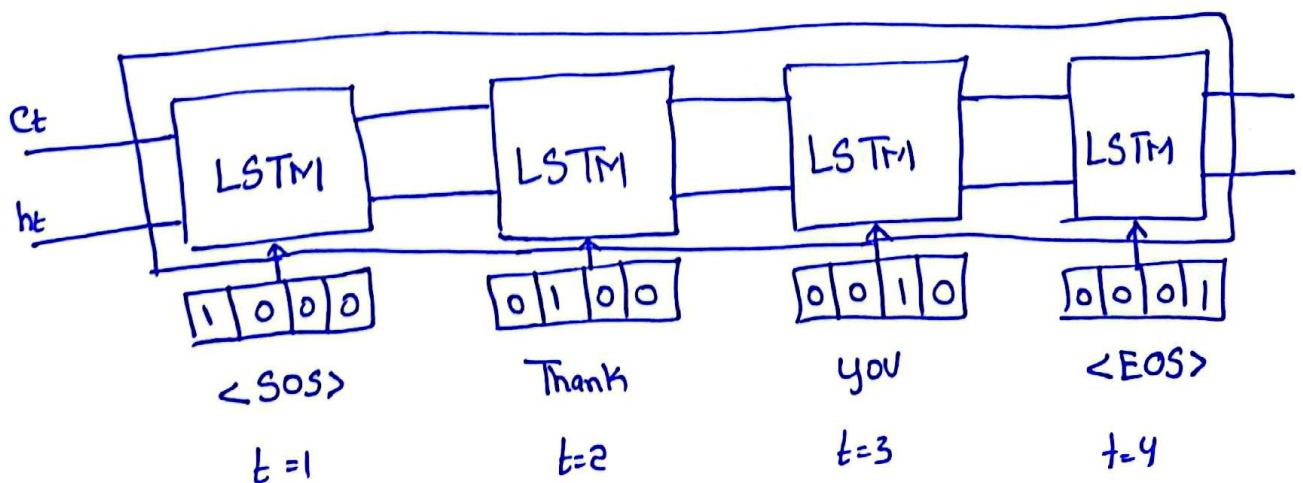
> **Use Cases**

① Language Translation

② Text Generation

③ Text Suggestion

→ **Sequence-to-Sequence (Seq2Seq)**

> **Encoder**

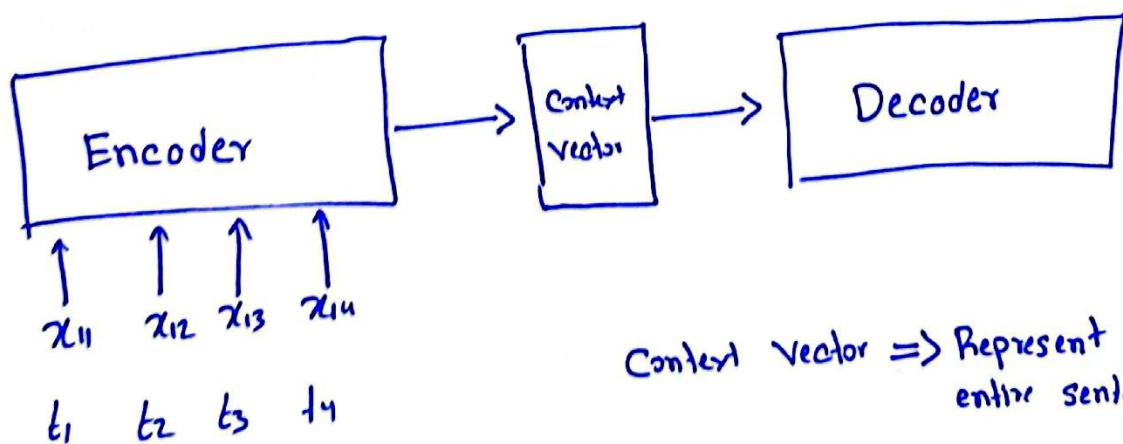• Text: Thank you

| C_t | LSTM | | LSTM | | LSTM | | LSTM |
| h_t | | | | | | | |

1 0 0 0		0 1 0 0		0 0 1 0		0 0 0 1
<SOS>		Thank		you		<EOS>
$t=1$		$t=2$		$t=3$		$t=4$

> Decoder

English: Thank you ——> French: Gracias

$[0.1, 0.6, 0.3]$ $[0.1, 0.3, 0.6]$

↑   ↑

| Softmax | | Softmax |

↑   ↑

| LSTM | —— | LSTM |

↑   ↑

| 1 | 0 | 0 | | 0 | 1 | 0 |

<SOS>   Gracias
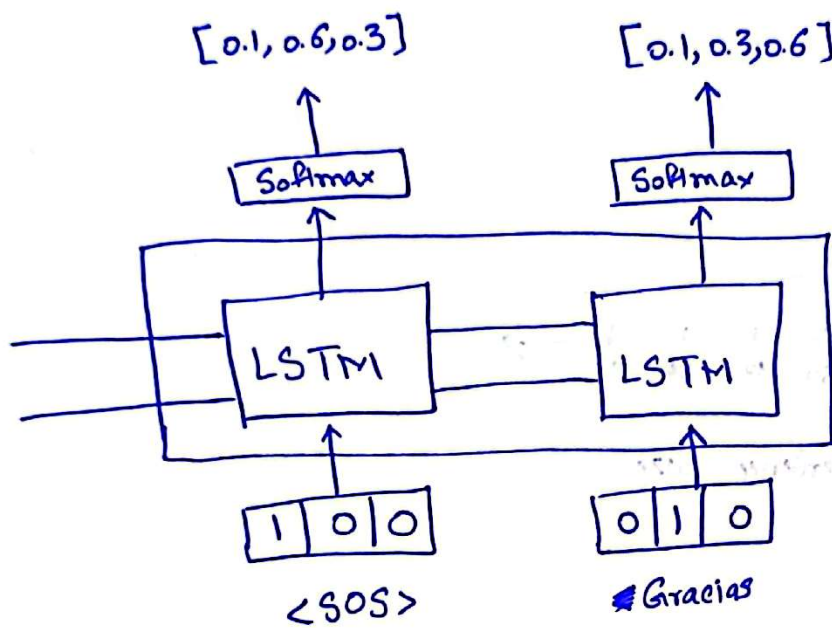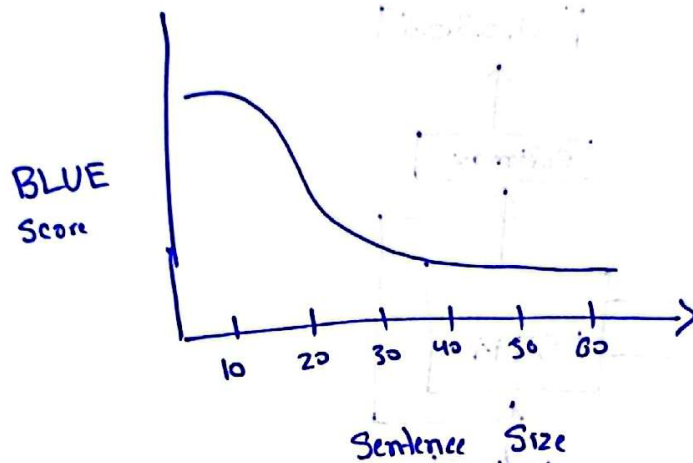
       t_1 t_2

y $[- \; - \; -]$ $[- \; - \; -]$

$\hat{y}$ $[- \; - \; -]$ $[- \; - \; -]$

——> Problems with Encoder Decoder Seq2Seq Architecture

| Encoder | ——> | Context Vector | ——> | Decoder |

↑   ↑   ↑   ↑

x_{11} x_{12} x_{13} x_{14}

t_1 t_2 t_3 t_4

Context   5   17   50   90
Size :

Context Vector => Represent entire sentence

Length of Sentence Varies

To solh
> Solution:

Attention Mechanism

Longer Paragraph ⟶ Context Vector
                              +
                    Additional Context